

3/9/2004


[> home](#) [> about](#) [> feedback](#) [> login](#)

US Patent & Trademark Office

Search Results

Search Results for: [(multiple or plural) <near> (operating <near> system) <near> exception <near> cisc <near> risc]

Found 32 of 127,944 searched.

Search within Results

[> Search Help/Tips](#)



[> Advanced Search](#)

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 32 short listing







Prev Page 1 2 Next Page

- 1 DISE: a programmable macro engine for customizing applications 30%


Marc L. Corliss , E. Christopher Lewis , Amir Roth
ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture May 2003
Volume 31 Issue 2
Dynamic Instruction Stream Editing (DISE) is a cooperative software-hardware scheme for efficiently adding customization functionality---e.g, safety/security checking, profiling, dynamic code decompression, and dynamic optimization---to an application. In DISE, application customization functions (ACFs) are formulated as rules for macro-expanding certain instructions into parameterized instruction sequences. The processor executes the rules on the fetched instructions, feeding the executi ...
- 2 Unconstrained speculative execution with predicated state buffering 24%

Hideki Ando , Chikako Nakanishi , Tetsuya Hara , Masao Nakaya
ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture May 1995
Volume 23 Issue 2
Speculative execution is execution of instructions before it is known whether these instructions should be executed. Compiler-based speculative execution has the potential to achieve both a high instruction per cycle rate and high clock rate. Pure compiler-based approaches, however, have greatly limited instruction scheduling due to a limited ability to handle side effects of speculative execution. Significant performance improvement is, thus, difficult in non-numerical applications. This paper ...
- 3 Programmed word length computer 22%


A. L. Lucke
Proceedings of the 1967 22nd national conference January 1967
The concept of a programmable word length computer has evolved through an attempt to minimize wasted storage in any fixed word length computer. Whether the computer contains six bits per word or 72 bits per word, storage is inevitably wasted when it is necessary to use a full word for a simple on-off switch, or to have 12-digit capacity used for 2-digit significance. Programmers have long recognized this. Thus, word packing and unpacking soon becomes a part of every programmer's repertoire. ...

- 4 Binary translation and architecture convergence issues for IBM system/390 8%
 Michael Gschwind , Kemal Ebcioglu , Erik Altman , Sumedh Sathaye
 Proceedings of the 14th international conference on Supercomputing May 2000
 We describe the design issues in an implementation of the ESA/390 architecture based on binary translation to a very long instruction word (VLIW) processor. During binary translation, complex ESA/390 instructions are decomposed into instruction "primitives" which are then scheduled onto a wide-issue machine. The aim is to achieve high instruction level parallelism due to the increased scheduling and optimization opportunities which can be exploited by binary translation software ...
- 5 An architectural framework for migration from CISC to higher performance 8%
 platforms
 Gabriel M. Silberman , Kemal Ebcioglu
 Proceedings of the 6th international conference on Supercomputing August 1992
 We describe a novel architectural framework that allows software applications written for a given Complex Instruction Set Computer (CISC) to migrate to a different, higher performance architecture, without a significant investment on the part of the application user or developer. The framework provides a hardware mechanism for seamless switching between two instruction sets, resulting in a machine that enhances application performance while keeping the same program behavior (from a user per ...
- 6 On structured data handling in parallel processing 6%
 Jean-Louis Lafitte
 ACM SIGARCH Computer Architecture News June 1995
 Volume 23 Issue 3
 The model we have developed at Universit  de Gen ve allows one to handle irregular data in parallel processing. It achieves much smoother data-driven parallel processing allowing far less discontinuities of the well-known pipeline "bubbles" type. This model is presented, along with some pre-liminary results.
- 7 Automatic derivation of compiler machine descriptions 6%
 ACM Transactions on Programming Languages and Systems (TOPLAS) July 2002
 Volume 24 Issue 4
 We describe a method designed to significantly reduce the effort required to retarget a compiler to a new architecture, while at the same time producing fast and effective compilers. The basic idea is to use the native C compiler at compiler construction time to discover architectural features of the new architecture. From this information a formal machine description is produced. Given this machine description, a native code-generator can be generated by a back-end generator such as BEG or burg ...
- 8 An out-of-order execution technique for runtime binary translators 5%
 Bich C. Le
 Proceedings of the eighth international conference on Architectural support for programming languages and operating systems October 1998
 Volume 32 , 33 Issue 5 , 11
 A dynamic translator emulates an instruction set architecture by translating source instructions to native code during execution. On statically-scheduled hardware, higher performance can potentially be achieved by reordering the translated instructions; however, this is a challenging transformation if the source architecture supports precise exception semantics, and the user-level program is allowed to register exception handlers. This paper presents a software technique which allows a translator ...
- 9 Usenet Nuggets 3%
 ACM SIGARCH Computer Architecture News March 1991
 Volume 19 Issue 1

10 A reduced register file for RISC architectures 2%


 Miquel Huguet , Tomás Lang
ACM SIGARCH Computer Architecture News September 1985
Volume 13 Issue 4

11 Multiple instruction issue in the NonStop cyclone processor 2%


 Robert W. Horst , Richard L. Harris , Robert L. Jardine
ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture May 1990
Volume 18 Issue 3

This paper describes the architecture for issuing multiple instructions per clock in the NonStop Cyclone Processor. Pairs of instructions are fetched and decoded by a dual two-stage prefetch pipeline and passed to a dual six-stage pipeline for execution. Dynamic branch prediction is used to reduce branch penalties. A unique microcode routine for each pair is stored in the large duplexed control store. The microcode controls parallel data paths optimized for executing the most frequent instr ...

12 Alpha AXP architecture 2%


 Richard L. Sites
Communications of the ACM February 1993
Volume 36 Issue 2

13 Shade: a fast instruction-set simulator for execution profiling 2%

 Bob Cmelik , David Keppel
ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1994 ACM SIGMETRICS conference on Measurement and modeling of computer systems May 1994
Volume 22 Issue 1


Tracing tools are used widely to help analyze, design, and tune both hardware and software systems. This paper describes a tool called Shade which combines efficient instruction-set simulation with a flexible, extensible trace generation capability. Efficiency is achieved by dynamically compiling and caching code to simulate and trace the application program. The user may control the extent of tracing in a variety of ways; arbitrarily detailed application state information may be collected ...

14 Security: Analyzing and modeling encryption overhead for sensor network nodes 1%

 Prasanth Ganesan , Ramnath Venugopalan , Pushkin Peddabachagari , Alexander Dean , Frank Mueller , Mihail Sichitiu
Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications September 2003

Recent research in sensor networks has raised security issues for small embedded devices. Security concerns are motivated by the deployment of a large number of sensory devices in the field. Limitations in processing power, battery life, communication bandwidth and memory constrain the applicability of existing cryptography standards for small embedded devices. A mismatch between wide arithmetic for security (32 bit word operations) and embedded data bus widths (often only 8 or 16 bits) combined ...

15 Embedded applications: Encryption overhead in embedded systems and sensor network nodes: modeling and analysis 1%

 Ramnath Venugopalan , Prasanth Ganesan , Pushkin Peddabachagari , Alexander Dean , Frank Mueller , Mihail Sichitiu
Proceedings of the international conference on Compilers, architectures and synthesis for embedded systems October 2003

Recent research in sensor networks has raised issues of security for small embedded devices. Security concerns are motivated by the deployment of a large number of sensory devices in the field. Limitations in processing power, battery life, communication bandwidth and memory constrain the applicability of existing cryptography standards for small embedded devices. A mismatch between wide arithmetic for security (32 bit word operations) and embedded data bus widths (often only 8 or 16 bits) combined ...

16 Retrospective: what have we learned from the PDP-11—what we have learned 1%



from VAX and Alpha

Gorden Bell , W. D. Strecker

25 years of the international symposia on Computer architecture (selected papers) August 1998

17 A cost-effective approach to implement a long instruction word microprocessor 1%



Yen-Jen Oyang , Bor-Ting Chang , Shu-May Lin

ACM SIGARCH Computer Architecture News March 1990

Volume 18 Issue 1

This paper presents a cost-effective approach to implement a long instruction word microprocessor. The proposed approach achieves the goal of optimal performance/cost tradeoff by selectively, instead of unconditionally, duplicating the hardware resources in a uniprocessor. A hardware resource is duplicated only if the performance improvement gained by doing so can justify the cost incurred. In accordance with this guideline, a long instruction word microprocessor architecture called the MPA arch ...

18 Sentinel scheduling: a model for compiler-controlled speculative execution 1%



Scott A. Mahlke , William Y. Chen , Roger A. Bringmann , Richard E. Hank , Wen-Mei W. Hwu , B. Ramakrishna Rau , Michael S. Schlansker

ACM Transactions on Computer Systems (TOCS) November 1993

Volume 11 Issue 4

Speculative execution is an important source of parallelism for VLIW and superscalar processors. A serious challenge with compiler-controlled speculative execution is to efficiently handle exceptions for speculative instructions. In this article, a set of architectural features and compile-time scheduling support collectively referred to as sentinel scheduling is introduced. Sentinel scheduling provides an effective framework for both compiler-controlled speculative executi ...

19 DAISY: dynamic compilation for 100% architectural compatibility 1%



Kemal Ebcioglu , Erik R. Altman

ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture May 1997

Volume 25 Issue 2

Although VLIW architectures offer the advantages of simplicity of design and high issue rates, a major impediment to their use is that they are not compatible with the existing software base. We describe new simple hardware features for a VLIW machine we call DAISY (Dynamically Architected Instruction Set from Yorktown). DAISY is specifically intended to emulate existing architectures, so that all existing softwa ...

20 Netwinder Office Server



0%



Jason Kroll

Linux Journal March 2000

Results 1 - 20 of 32 short listing

 
Prev Page 1 2 Next Page

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.